IN THE CLAIMS:

Please amend the claims as follows:

1. (Previously Amended) A method for buffering data produced by a computer graphics pipeline, comprising:
producing graphics floating point data in a graphics pipeline;
operating on the graphics floating point data in the graphics pipeline; and
storing the graphics floating point data to a buffer;
wherein the graphics floating point data includes fragment data received from a rasterizer that is read and stored in an unclamped format dictated by a graphics application program interface for increasing a parameter selected from the group consisting of a precision and a range of the graphics floating point data.

2.-3. (Cancelled)

4. (Previously Amended) The method as recited in claim 1, wherein the fragment data includes color data.

5. (Previously Amended) The method as recited in claim 1, wherein the fragment data includes depth data.

6. (Original) The method as recited in claim 1, wherein the graphics floating point data is only constrained by an underlying data type.

7. (Original) The method as recited in claim 1, wherein the buffer serves as a texture map.

8. (Previously Amended) A computer program product for buffering data produced by a computer graphics pipeline, comprising:

(a) computer code for producing graphics floating point data in a graphics pipeline;

(b) computer code for operating on the graphics floating point data in the graphics pipeline; and

(c) computer code for storing the graphics floating point data to a buffer;

(d) wherein the graphics floating point data includes fragment data received from a rasterizer that is read and stored in an unclamped format dictated by a graphics application program interface for increasing a parameter selected from the group consisting of a precision and a range of the graphics floating point data.

9.-10. (Canceled)

11. (Previously Amended) The computer program product as recited in claim 8, wherein the fragment data includes color data.

12. (Previously Amended) The computer program product as recited in claim 8, wherein the fragment data includes depth data.

13. (Original) The computer program product as recited in claim 8, wherein the graphics floating point data is only constrained by an underlying data type.

14. (Original) The computer program product as recited in claim 8, wherein the buffer serves as a texture map.

15. (Previously Amended) A system for buffering data produced by a computer graphics pipeline, comprising:

(a) logic for producing graphics floating point data in a graphics pipeline;

(b) logic for operating on the graphics floating point data in the graphics pipeline; and

(c) logic for storing the graphics floating point data to a buffer;

(d) wherein the graphics floating point data includes fragment data received from a rasterizer that is read and stored in an unclamped format dictated by a graphics application program interface for increasing a parameter selected from the group consisting of a precision and a range of the graphics floating point data.

16. (Previously Amended) A buffering apparatus, comprising:

(a) a buffer capable of storing graphics floating point data produced by a graphics pipeline;

(b) wherein the graphics floating point data includes fragment data received from a rasterizer that is stored in an unclamped format dictated by a graphics application program interface for increasing a parameter selected from the group consisting of a precision and a range of the graphics floating point data.

17. (Previously Amended) A system for buffering data produced by a computer graphics pipeline, comprising:

(a) means for producing graphics floating point data in a graphics pipeline;

(b) means for operating on the graphics floating point data in the graphics pipeline; and

(c) means for storing the graphics floating point data to a buffer;

(d) wherein the graphics floating point data includes fragment data received from a rasterizer that is read and stored in an unclamped format dictated by a graphics application program interface for increasing a parameter selected from the group consisting of a precision and a range of the graphics floating point data.

18. (Cancelled)

19. (Cancelled)

20. (Cancelled)

21. (Cancelled)

22. (Cancelled)

23. (Cancelled)

24. (Previously Amended) A method for buffering data produced by a computer graphics pipeline, comprising:
operating on graphics floating point data in a graphics pipeline;
producing the graphics floating point data in the graphics pipeline;
determining whether the graphics pipeline is operating in a programmable mode utilizing a command associated with a graphics application program interface;
if it is determined that the graphics pipeline is not operating in the programmable mode, performing standard graphics application program interface operations on the graphics floating point data; and
if it is determined that the graphics pipeline is operating in the programmable mode:
storing the graphics floating point data to a frame buffer,
wherein the graphics floating point data includes fragment data received from a rasterizer that is read and stored in an unclamped format dictated by a graphics application program interface extension for increasing a parameter selected from the group consisting of a precision and a range of the graphics floating point data.

25. (Currently Amended) ~~The buffering apparatus as recited in claim 19,~~ A buffering apparatus, comprising:
a buffer capable of storing graphics floating point data produced by a graphics pipeline;
wherein the buffer serves as a texture map;

wherein the buffer serves as the texture map by using previous rendering results via an extension of an application program interface.

26. (Previously Presented) The method as recited in claim 1, wherein the buffer includes a frame buffer.